domogik-plugin-generic

Release 1.0

December 27, 2016

Contents

CHAPTER 1

Plugin generic

1.1 Purpose

The generic plugin is used to handle xPL messages sent by xPL clients that are not part of Domogik. They can be *arduino* DIY devices, ... The generic plugin contains only the description of the xPL messages to catch and so there is nothing to start/stop and nothing to configure.

You just have to install the plugin, check it is present in the clients list and create some devices.

Please notice that you must see your external xPL clients in the clients list as **xPL clients**. If you don't see them, it means that the xPL hub doesn't see them and so the plugin won't be able to catch the xPL messages.

To be detected by the xPL hub (and Domogik), a xPL client must implement the heartbeat xPL messages. More informations here : http://xplproject.org.uk/wiki/index.php/XPL_Specification_Document#Heartbeat_Messages Notice that some DIY projects don't implement the heartbeat messages and so Domogik won't see them!

If you want to **create your own arduino xPL client**, you can use this library : https://github.com/olebrun/xPL.Arduino

You can check if a xPL message is received on Domogik side by launching the **dmg_dump** tool.

1.2 Dependencies

There is no dependency.

1.3 Plugin configuration

There is nothing to configure.

1.4 Create the domogik devices

- Temperature devices
- Humidity devices
- CO2 devices
- Solar exposure (insolation) devices
- Pressure devices

- Total volatile organic compound devices
- RGB devices

1.5 Set up your widgets on the user interface

You can now place the widgets of your devices features on the user interface.

CHAPTER 2

Temperature devices

2.1 Overview

These devices are able to send the temperature.

2.2 Configuration

When you create such a device, you will have to set this parameter:

Key	Type	Description
device	string	The device address.

2.3 xPL message

Here is a sample of a compliant xPL message:

```
xpl-trig
{
hop=1
source=arduino-temp.myarduino
target=*
}
sensor.basic
{
current=13
type=temp
device=tempsensor1
}
```

- **current** : contains the value
- **device** : contains the device address

CHAPTER 3

Humidity devices

3.1 Overview

These devices are able to send the humidity.

3.2 Configuration

When you create such a device, you will have to set this parameter:

Key	Type	Description
device	string	The device address.

3.3 xPL message

Here is a sample of a compliant xPL message:

```
xpl-trig
{
hop=1
source=arduino-humidity.myarduino
target=*
}
sensor.basic
{
current=75
type=temp
device=humiditysensor1
}
```

- **current** : contains the value
- **device** : contains the device address

CHAPTER 4

Pressure devices

4.1 Overview

These devices are able to send the pressure in Pascal.

4.2 Configuration

When you create such a device, you will have to set this parameter:

Key	Type	Description
device	string	The device address.

4.3 xPL message

Here is a sample of a compliant xPL message:

```
xpl-trig
{
hop=1
source=arduino-pressure.myarduino
target=*
}
sensor.basic
{
current=34
type=pressure
device=pressuresensor1
}
```

- **current** : contains the value
- **device** : contains the device address

CHAPTER 5

CO2 devices

5.1 Overview

These devices are able to send the carbon dioxide informations.

5.2 Configuration

When you create such a device, you will have to set this parameter:

Key	Type	Description
device	string	The device address.

5.3 xPL message

Here is a sample of a compliant xPL message:

```
xpl-trig
{
hop=1
source=arduino-co2.myarduino
target=*
}
sensor.basic
{
current=456
type=co2
device=co2sensor1
}
```

- **current** : contains the value
- **device** : contains the device address

CHAPTER 6

Solar exposure (insolation) devices

6.1 Overview

These devices are able to send the solar exposure (insolation).

6.2 Configuration

When you create such a device, you will have to set this parameter:

Key	Type	Description
device	string	The device address.

6.3 xPL message

Here is a sample of a compliant xPL message:

```
xpl-trig
{
hop=1
source=arduino-insolation.myarduino
target=*
}
sensor.basic
{
current=123
type=insolation
device=insolationsensor1
}
```

- **current** : contains the value
- **device** : contains the device address

CHAPTER 7

Total volatile organic compound devices

7.1 Overview

These devices are able to send the total volatile organic compound.

7.2 Configuration

When you create such a device, you will have to set this parameter:

Key	Type	Description
device	string	The device address.

7.3 xPL message

Here is a sample of a compliant xPL message:

```
xpl-trig
{
hop=1
source=arduino-tvoc.myarduino
target=*
}
sensor.basic
{
current=752
type=tvoc
device=tvocsensor1
}
```

- **current** : contains the value
- **device** : contains the device address

CHAPTER 8

RGB devices

8.1 Overview

These devices are able to control RGB devices.

8.2 Configuration

When you create such a device, you will have to set this parameter:

Key	Type	Description
device	string	The device address.

8.3 xPL message

8.3.1 xpl-trig/xpl-stat

Here is a sample of a compliant xPL message:

```
xpl-trig
{
hop=1
source=arduino-rgb.myarduino
target=*
}
arduino.rgb
{
command=setcolor
color=ff0000
device=rgb
}
```

- **color** : contains the color value in hexa RGB format without the #. Example for red : 'ff0000', for blue : '0000ff'
- **device** : contains the device address

8.3.2 xpl-cmnd

Here is a sample of a compliant xPL message:

```
xpl-cmnd
{
hop=1
source=arduino-rgb.myarduino
target=*
}
arduino.rgb
{
command=setcolor
color=ff0000
device=rgb
}
```

- **color** : contains the color value to set in hexa RGB format without the #. Example for red : 'ff0000', for blue : '0000ff'
- **device** : contains the device address

CHAPTER 9

Development informations

There is no python part for this plugin. All is handled in the *info.json* file.

CHAPTER 10

Tests

There is no automated tests for this plugin.

10.1 Manual tests

10.1.1 Create the devices

In Domogik administration, create the following devices :

Device type	Device name	Device address
RGB device	rgb	parentsbedroom
CO2 sensors	co2	co2sensor1
Humidity sensors	humidity	humiditysensor1
Insolation sensors	insolation	insolationsensor1
Pressure sensors	pressure	pressuresensor1
Temperature sensors	temperature	tempsensor1
Tvoc sensors	tvoc	tvocsensor1

10.1.2 Simulate some xPL clients

Once all device are created, launch the following commands :

```
cd tests
python co2.py
python humidity.py
python insolation.py
python pressure.py
python rgb.py
python temperature.py
python tvoc.py
```

10.1.3 Check the results

In Domogik administration, for each device, display its details and check the sensor value.

The expected values are :

Device name	Value
rgb	00ff00
co2	213
humidity	75
insolation	1050
pressure	75
temperature	13
tvoc	45

CHAPTER 11

Changelog

11.1 1.1

- New sensor and command : RGB devices
- New sensor : humidity
- New sensor : pressure
- New sensor : co2
- New sensor : solar exposure
- New sensor : total volatile organic compound
- Experimental sensor and command : OSD

11.2 1.0

- Plugin creation
- Add sensor : temperature